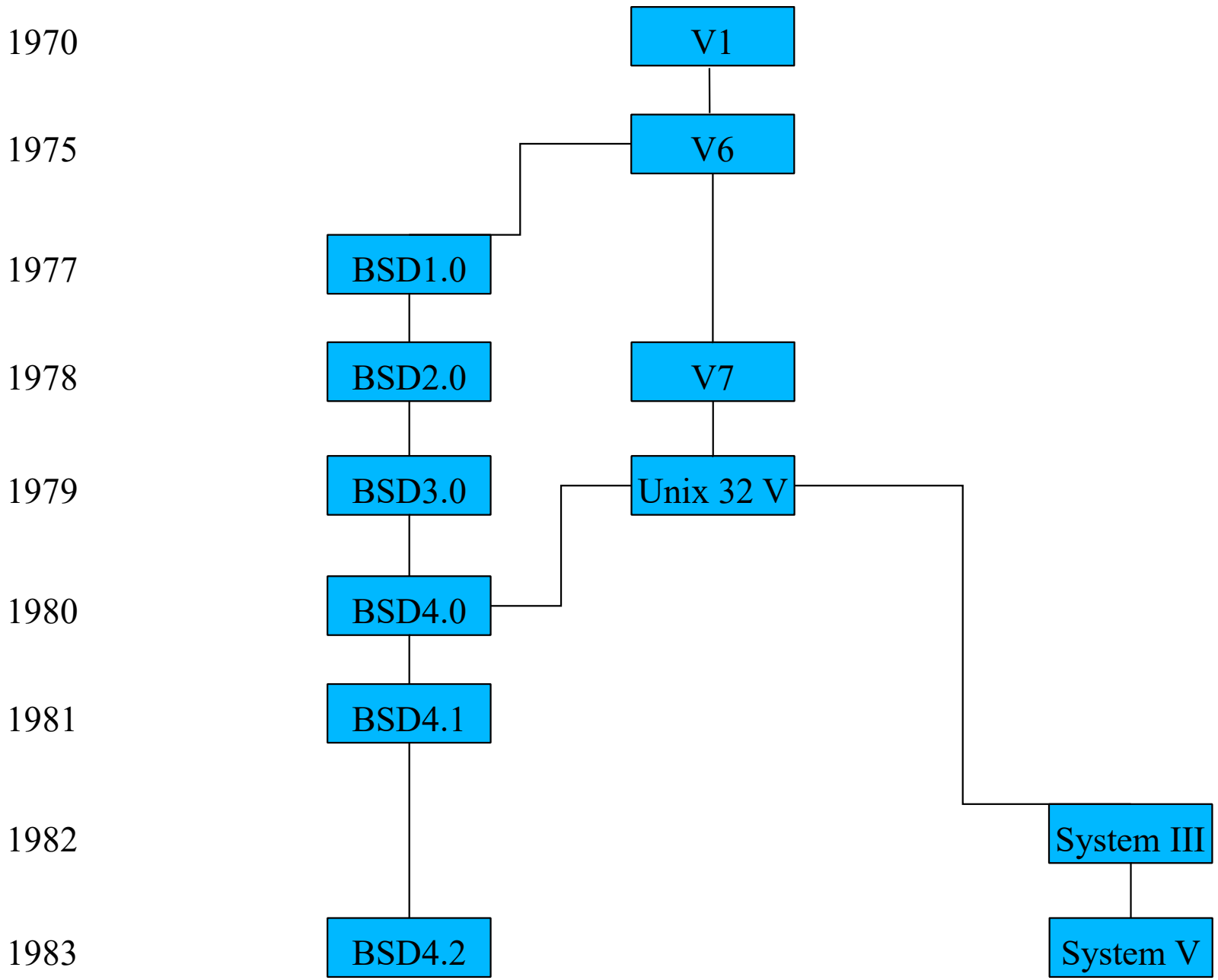
HỆ ĐIỀU HÀNH UNIX

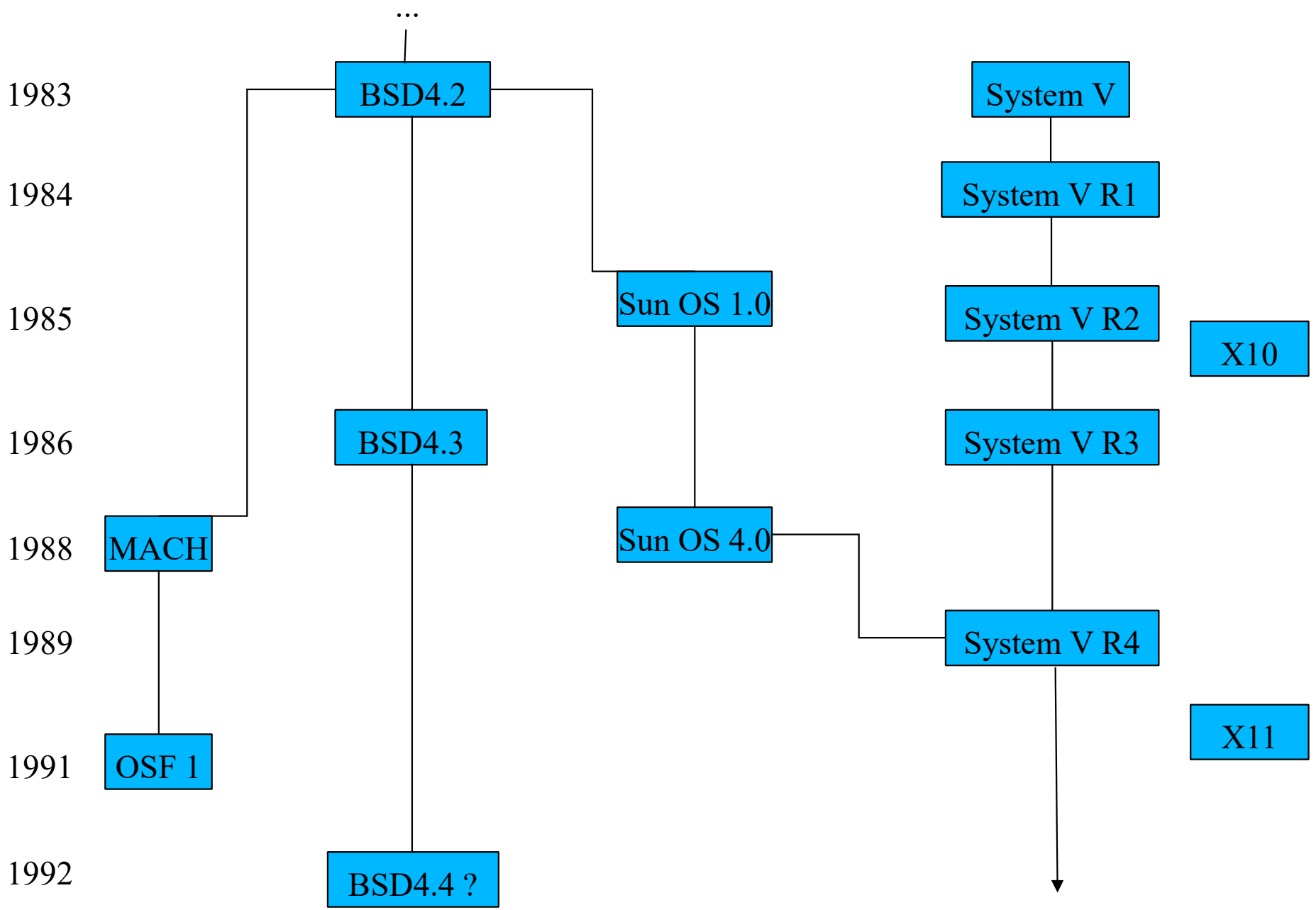
GIỚI THIỆU
HỆ ĐIỀU HÀNH UNIX

1. Tổng quan
2. Các thành phần hệ điều hành UNIX
3. Một sô thao tác cơ bản

Lịch sử phát triển Unix

- 1969: Thiết kế phiên bản đầu tiên bởi Ken Thompson trong phong thí nghiệm Bell Lab của AT&T
- 1973: Viết lại bằng ngôn ngữ C để cho phép cài đặt UNIX trên nhiều hệ thống khác nhau
- 1975: Phân phối sản phẩm V6 trong các trường đại học
- 1977: Xuất hiện phiên bản Unix đầu tiên dùng trong các trường đại học, BSD (Berkeley Software Distribution)
- 1978 : Phân phối V7 trong lĩnh vực công nghiệp
- 1984 : Ra đời X-Window (X11) trong Unix
- 1990 : Ra đời chuẩn POSIX cho thư viện của UNIX
- Ngày nay UNIX là hệ thống mở phát triển xung quanh một hạt nhân POSIX, các tiện ích, các môi trường hệ thống, giao diện đồ hoạ,...


1970
1975
1977
1978
1979
1980
1981
1982
1983
V1
V6
BSD1.0
BSD2.0
V7
BSD3.0
Unix 32 V
BSD4.0
BSD4.1
System III
BSD4.2
System V


...
1983
1984
1985
1986
1988
1989
1991
1992
BSD4.2
BSD4.3
BSD4.4 ?
MACH
OSF 1
Sun OS 1.0
Sun OS 4.0
System V
System V R1
System V R2
System V R3
System V R4
X10
X11

ĐẶC ĐIỂM
HỆ ĐIỀU HÀNH UNIX

UNIX

- là một hệ điều hành
  - đa nhiệm
  - đa người sử dụng
  - có tính mô đun hoá cao
  - không phụ thuộc vào phần cứng
  - hỗ trợ môi trường phát triển ứng dụng

Tính đa nhiệm

- Một chương trình khi chạy trong máy tính là một tiến trình
  - đa nhiệm có nghĩa là nhiều tiến trình có thể chạy cùng một thời điểm
  - tiến trình không phải là chương trình
  - có thể chạy nhiều tiến trình cho cùng một chương trình tại một thời điểm
  - Hệ điều hành nào là đa nhiệm: DOS, NT, Windows 9x, Windows 2000, Windows XP ?

Tính đa người sử dụng

- Nhiều người sử dụng có thể cùng truy xuất vào hệ thống tại một thời điểm
  - cần có khái niệm tài khoản sử dụng nhưng có nhiều tài khoản không đồng nghĩa với đa người sử dụng
  - một tiến trình tạo ra thuộc quyền sở hữu người đã tạo ra nó
  - do đó các tiến trình có thể thuộc quyền sở hữu của nhiều người khác nhau
  - Hệ điều hành nào là đa người sử dụng: DOS, NT, Windows 9x, Windows 2000, Windows XP ?

Tính mô đun

- Mô đun hoá về kiến trúc
  - Hạt nhân quản lý các nhiệm vụ ở mức thấp
  - Tầng ứng dụng cung cấp các tiện ích sử dụng đối với người sử dụng
- Mô đun hoá về ứng dụng
  - Cung cấp nhiều công cụ nhỏ, chuyên dụng nhưng đa dạng để hỗ trợ công việc người sử dụng
  - Không cung cấp các công cụ có tính đa năng nhưng người sử dụng làm được rất nhiều việc phức tạp bằng cách kết hợp các công cụ nhỏ với nhau

GNU/LINUX (1)

- 1984 : khởi xướng dự án GNU bởi Richard Stallman với mục đích phát triển một hệ điều hành đầy đủ, tựa Unix nhưng có mã nguồn mở
  - GNU cho ra đời nhiều tiện ích UNIX được sử dụng ngày nay : emacs, gcc,…
  - Vẫn cần phải phát triển một hạt nhân để có một hệ điều hành đầy đủ
- 1991 : Linus Torvald đã công bố phiên bản LINUX đầu tiên, một hạt nhân UNIX, đồng thời yêu cầu hỗ trợ phát triển của cộng đồng lập trình viên

GNU/LINUX (2)

- Sự kết hợp giữa hạt nhân LINUX và các tiện ích GNU đã cho ra đời một hệ điều hành GNU/ LINUX đầy đủ, có sức mạnh và miễn phí cho rất nhiều dòng máy tính khác nhau
  - Intel x86, Alpha, ARM, Power PC (Macintosh), PDA
- Chú ý
  - Tên LINUX vẫn thường được dùng cho cả hệ điều hành bao gồm hạt nhân Linux và các tiện ích kèm theo

Hạt nhân Linux (1)

- 1991 : Phiên bản đầu tiên (version 0.01).
- 1992 : Phiên bản 0.96 có rất nhiều chức năng và sở hữu một giao diện đồ hoạ X Window (Xfre86)
- 1993 : Có hơn 100 lập trình viên tham gia phát triển Linux (version 0.99)
- 1994 : Ra đời phiên bản 1.0. Cách đánh số các phiên bản tuân thủ theo nguyên tắc:
  - <phiên bản chính>.<phiên bản phụ>.<phiên bản gỡ lỗi>
  - Các phiên bản có số phụ giống nhau thì không có chức năng mới
  - Tất cả các phiên bản ổn định đều có số phụ là chẵn
  - Các phiên bản "bêta" khi thêm chức năng đều có số phụ là lẻ

Hạt nhân Linux (2)

- 1996: Ra đời Linux 2.0 và được sự dụng trong công nghiệp
- 1997: Xuất hiện các tạp chí chuyên đề về Linux ở nhiều nước trên thế giới
- 2001: Ra đời phiên bản 2.4. Đây là hạt nhân có tính ổn định và được sử dụng trong hầu hết các bản phân phối Linux
- Ngày nay: Hạt nhân Linux đang đi vào giai đoạn cuối. Người lập trình không đưa thêm vào các chức năng mới nữa mà tập trung vào gỡ lỗi và tạo ra phiên bản ổn định nhất

Đặc điểm của HĐH LINUX

- Tương thích với chuẩn POSIX, System V và BSD
- Hỗ trợ giả lập thiết bị đầu cuối
- Hỗ trợ các console ảo
- Có thể cài đặt với các HĐH khác (dùng LILO, GNUB)
- Đọc được dữ liệu trên nhiều định dạng lưu trữ: etx2fs, ms-dos, vfat, iso 9660,…
- Cài đặt đầy đủ các chuẩn giao thức mạng: TCP/IP, SLIP, PPP, NFS,…
- Giao diện đồ hoạ: X Window KDE & Gnome
- Hỗ trợ rất nhiều dịch vụ ứng dụng: CSDL, ƯD văn phòng, dịch vụ internet, …
- Hỗ trợ tích hợp mạng với các HĐH khác như Windows

Linux với Window

- Tại sao dùng LINUX?
  - đủ tin cậy để đảm bảo HĐH có thể thực nhiều công việc nặng
  - Tốt hơn nhiều Windows trong khía cạnh quản lý công việc và quản lý mạng
  - Ít lỗi hệ thống và chạy ổn đinh hơn nhiều so với Windows
  - Miễn phí nhưng rất đầy đủ
  - Là lựa chọn tuyệt vời trong giảng dạy và nghiên cứu

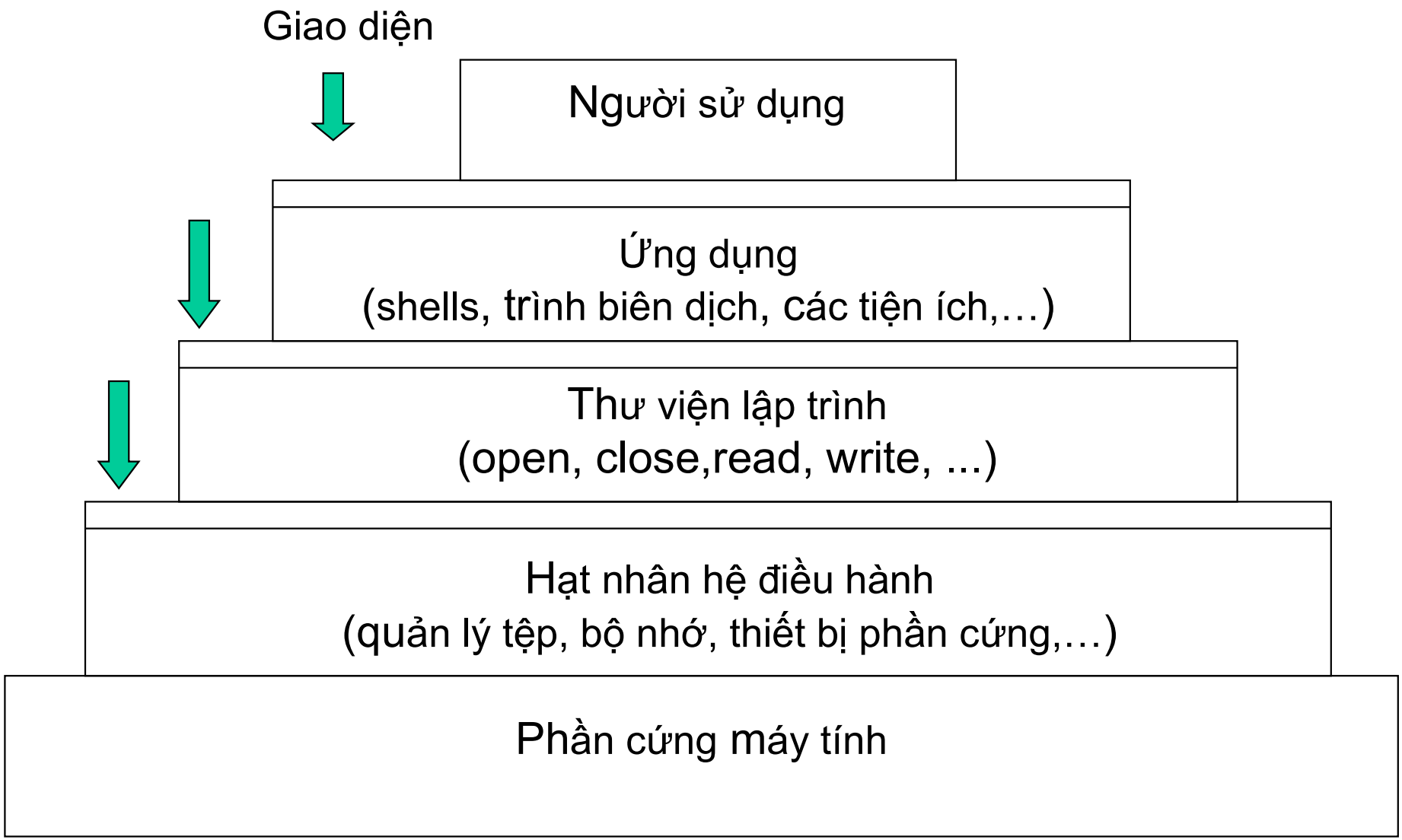
GIỚI THIỆU
HỆ ĐIỀU HÀNH UNIX

1. Tổng quan
2. Các thành phần hệ điều hành UNIX
3. Một sô thao tác cơ bản

THÀNH PHẦN CỦA HỆ ĐIỀU HÀNH UNIX

- Kernel
- Shell

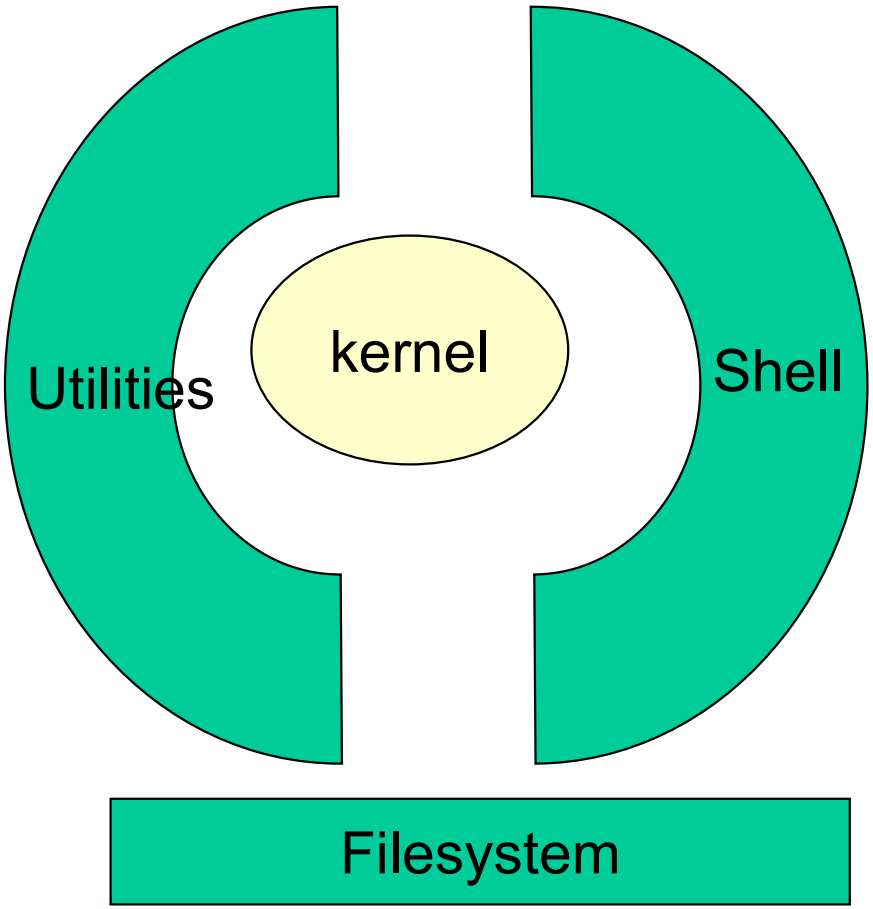
Kiến trúc hệ thống


Giao diện
Người sử dụng
Ứng dụng
(shells, trình biên dịch, các tiện ích,…)
Thư viện lập trình
(open, close,read, write, ...)
Hạt nhân hệ điều hành
(quản lý tệp, bộ nhớ, thiết bị phần cứng,…)
Phần cứng máy tính

KERNEL

Nhân của UNIX

- được nạp tại thời điểm khởi động và thiết lập hệ thống,
  - –tạo các tiến trình hệ thống ban đầu.
  - –duy trì trong bộ nhớ và quản trị hệ thống
- *Quản trị tài nguyên*/mediator -
  - –khái niệm rất cơ bản
  - –Chia sẻ thời gian CPU,
  - –cộng tác truy nhập đến các ngoại vi,
  - –quản trị bộ nhớ ảo.
  - –Thực thi các thao tác *đặc quyền(privileged)*.
- cung cấp các hàm nguyên thủy đồng bộ.
- Định nghĩa các điểm vào(*entry points*)
  - –*syscall*, *exceptions* hay *interrupts*.



Kernel là trái tim của OS – quản trị phần cứng

Shell là giao tiếp dòng lệnh với người dùng, cho phép tương tác với kernel

Utilities là các chương trình có nhiệm vụ chung như sao chép, di chuyển, đổi tên hay xóa file

Nhân Unix

Nhân Unix tập hợp các đặc trưng phần cứng

Proceses (time sharing, protected addres space)

Signals
and semaphores

kernel

Virtual memory
(swapping, paging, mapping)

The **filesystem**
(files, directories, namespace)

Interproces communication
(pipes and network connections)

Các tiến trình

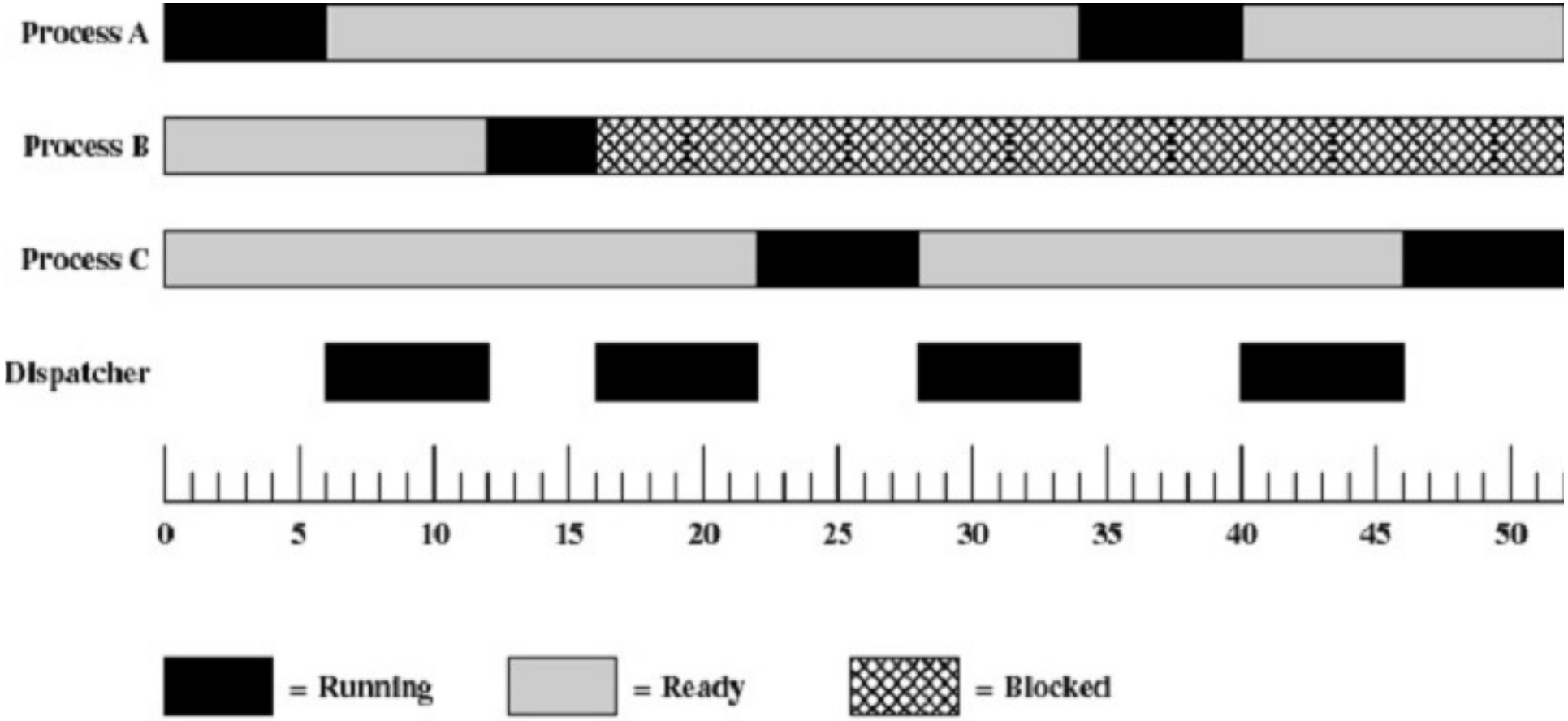
- *Một tiến trình là một chương trình đang vận hành; sự vận hành tiến trình phải diễn ra theo phương thức nối tiếp. Có thể đặc trưng qua dấu vết của nó.*
- Một tiến trình yêu cầu tài nguyên được hệ điều hành quản trị.
- OS xen kẽ các vận hành của một số tiến trình nhằm tăng hiệu suất sử dụng đến mức tối đa
- OS hỗ trợ truyền thông giữa các tiến trình(***IPC***) và tạo các tiến trình người sử dụng

Chế độ, không gian và ngữ cảnh

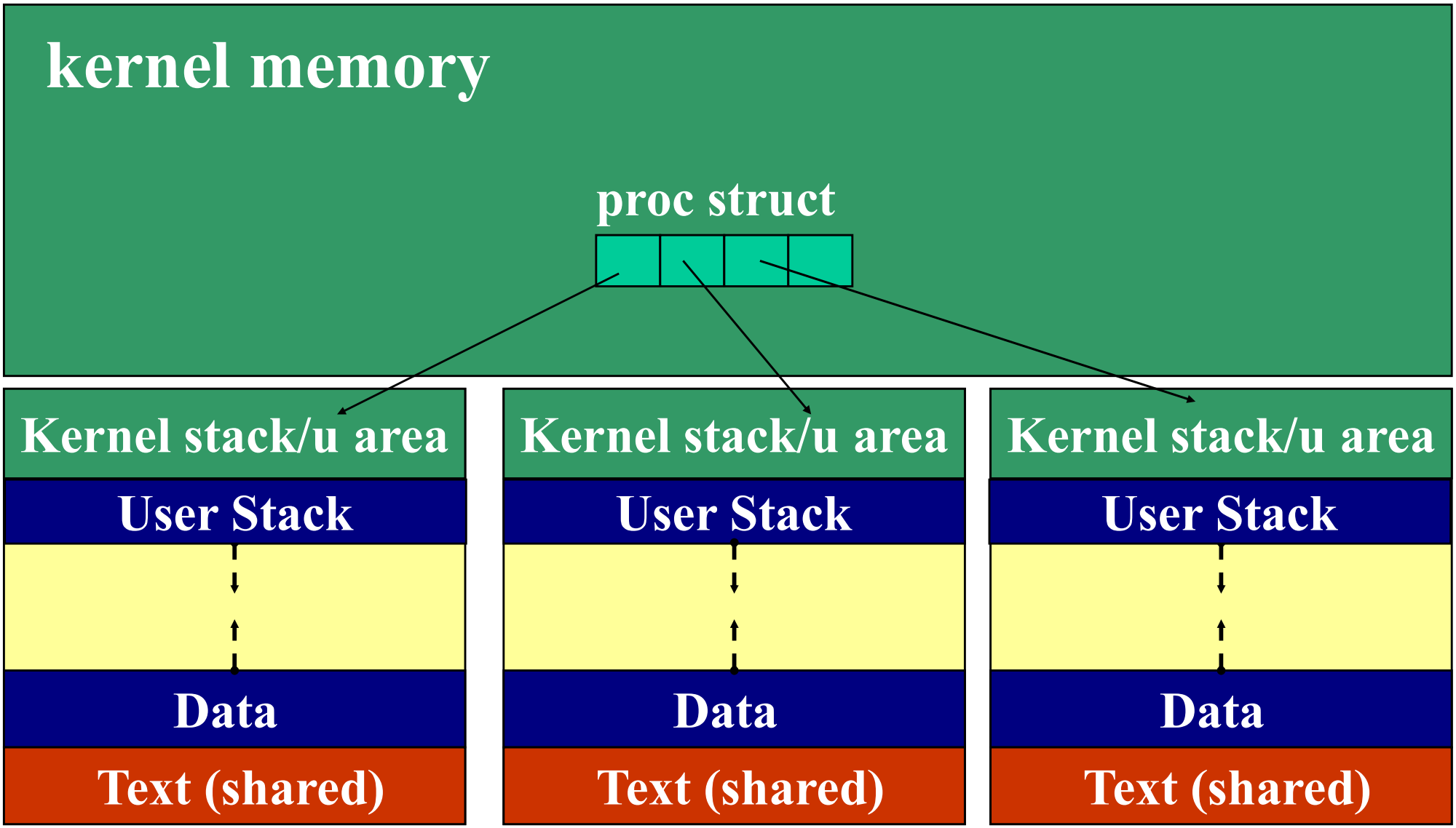
		Privileged
mode / ***context***	**user**	**kernel**
process	***Application (user code)***	***System calls Exceptions***
kernel	**X not allowed**	***Interrupts, System tasks***

system space

Chia sẻ thời gian các tiến trình

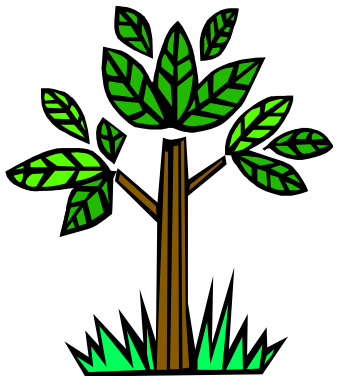


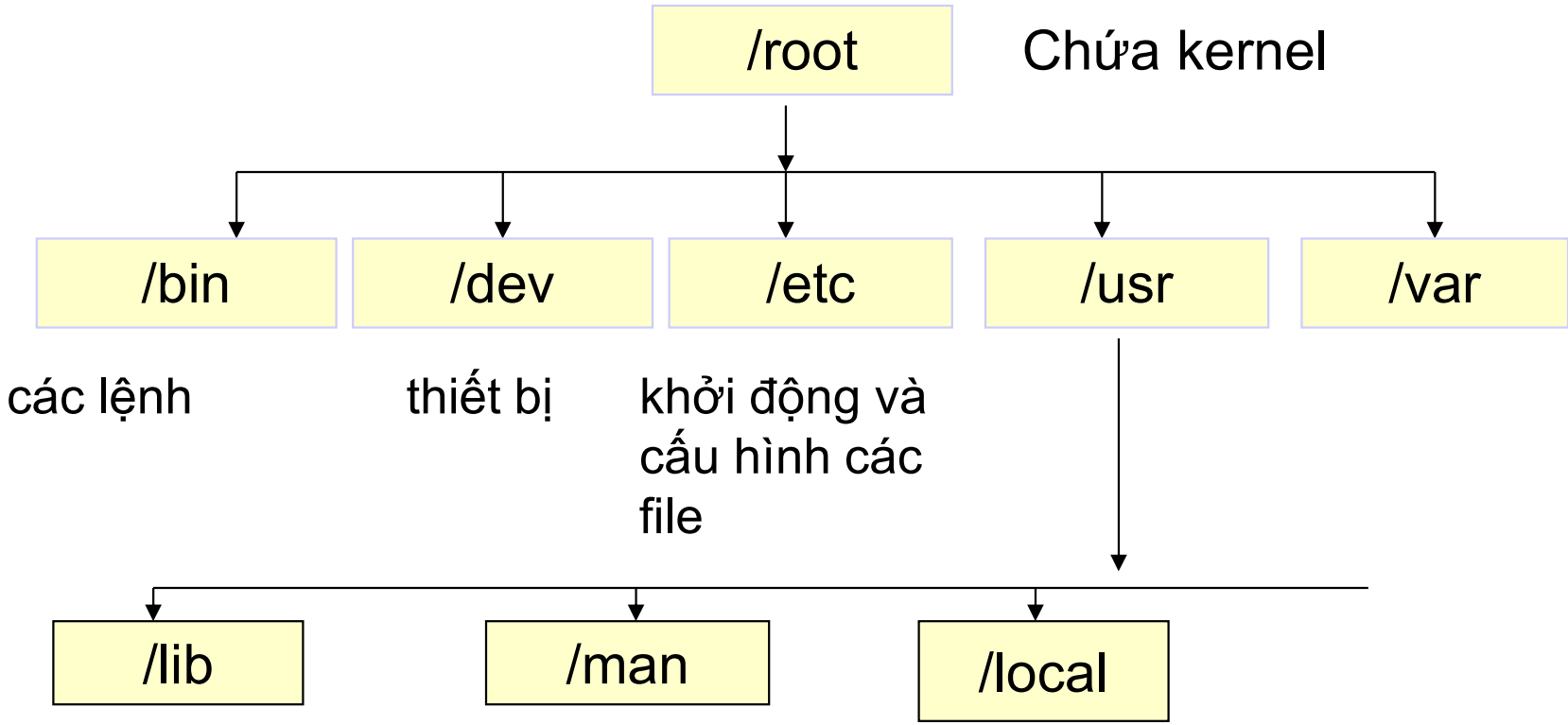
Chia sẻ bộ nhớ



Hệ thống file (Filesystem)

Hầu hết các hệ thống file đều là phân chương đĩa





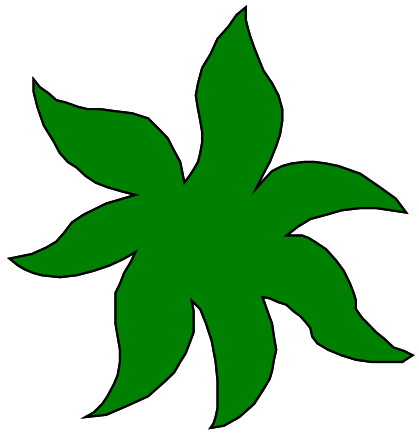
Các kiểu File

Thường(Regular Files)

Thư mục
(Directory)

File thiết bị Character

File thiết bị Block

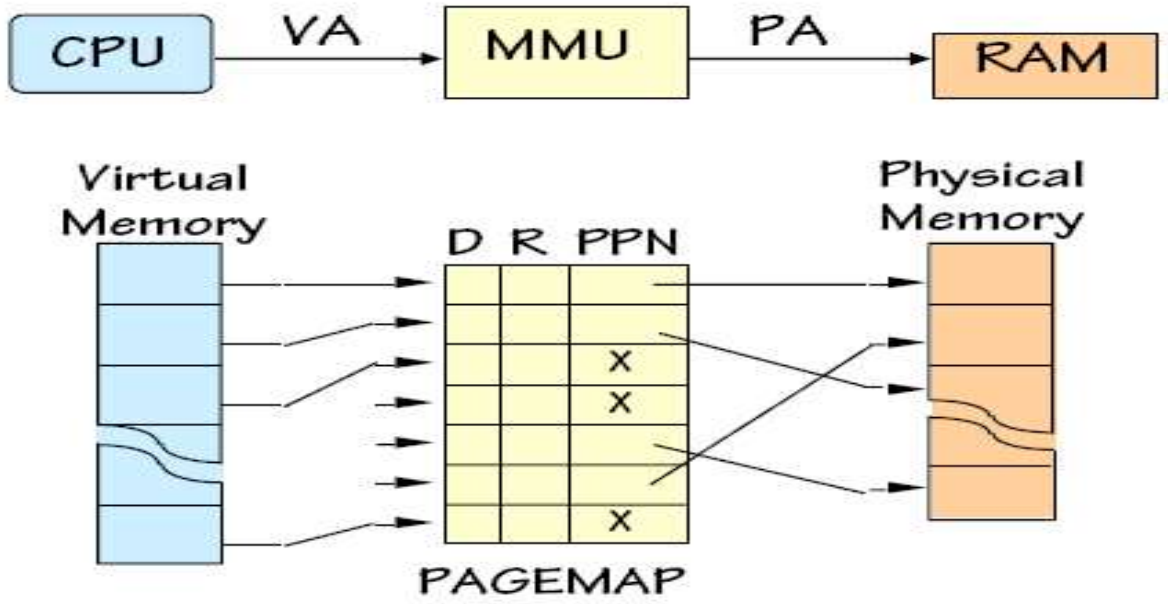


UNIX domain sockets

Đường ống (FIFO)

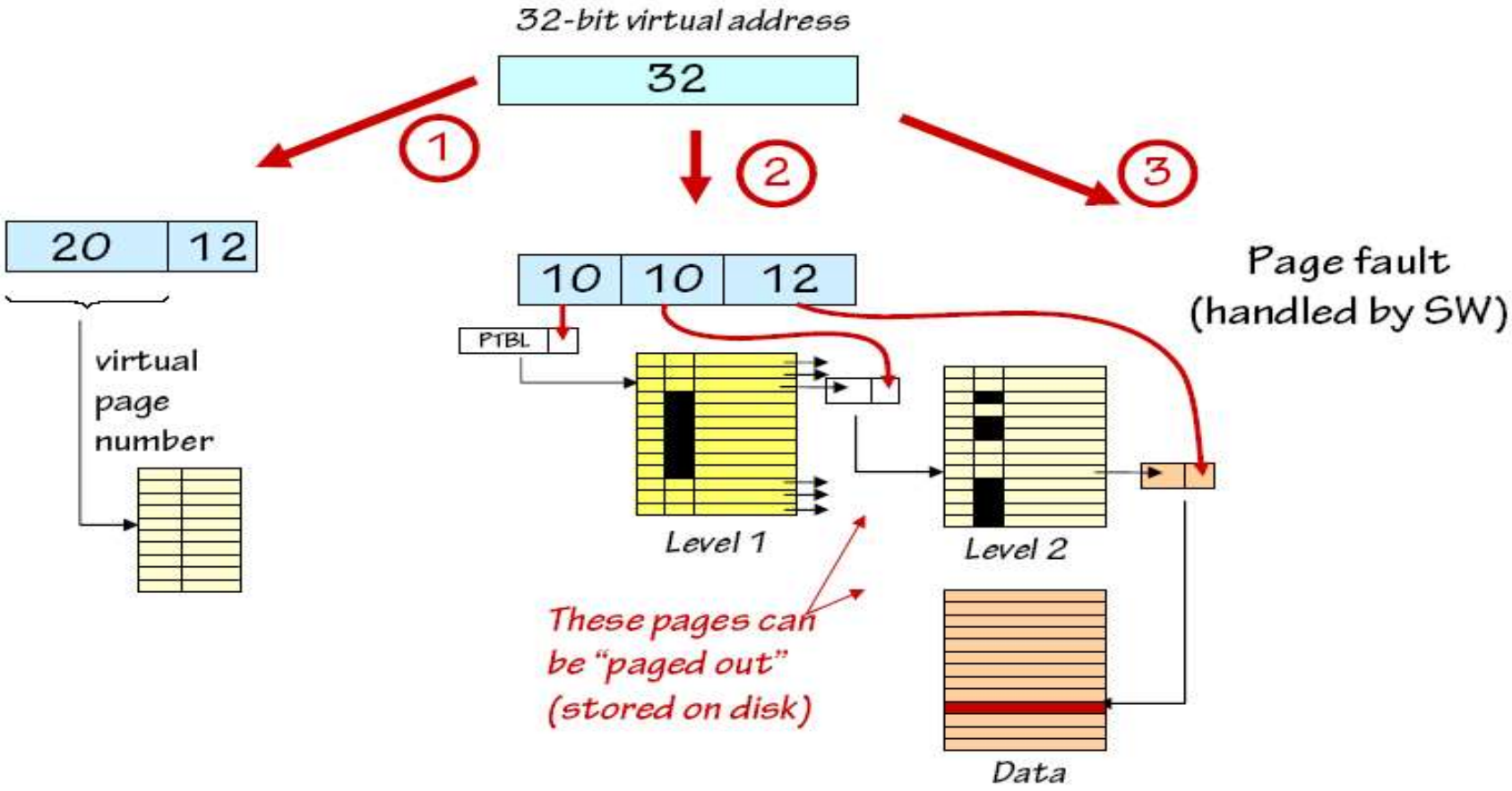
Các liên kết biểu
tượng(Symbolic)

Bộ nhớ ảo



- **Mục tiêu: tạo một cảm giác không gian địa chỉ ảo lớn**
- chia địa chỉ thành **(VPN,ofset), ánh xạ đến (PPN,ofset)**
- sử dụng các bit địa chỉ cao để chọn trang:giữ các dữ liệu quan hệ nhau trên cùng một trang
- sử dụng **cache (TLB-Translation Lookaside Buffer) để tăng tốc cơ chế ánh xạ**
- độ trễ đĩa lớn: giữ tập làm việc trong bộ nhớ vật lý, sử dụng **write-back**

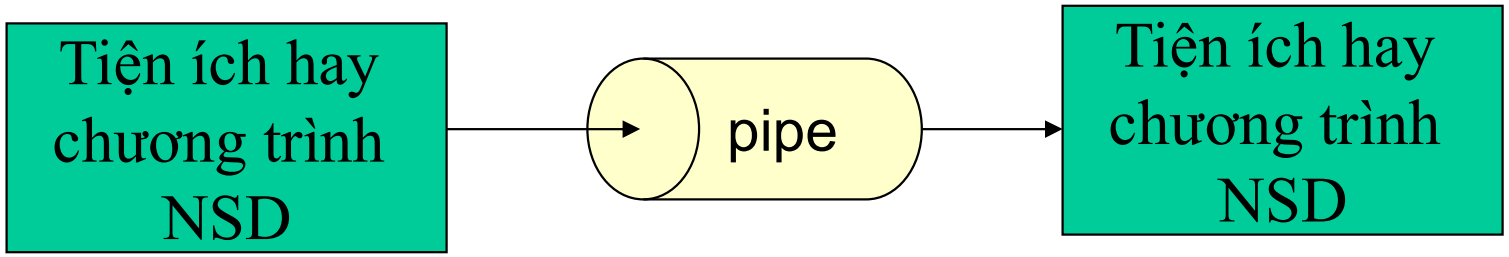
Dịch địa chỉ- Cách tiếp cận nhiều mức



- Nhìn vào bảng TLB: VPN→PPN cache
- Thường thực thi như một lối vào nhỏ (16-64) liên kết đầy đủ

Named Pipes

Pipelining cho phép ta kết nối hai chương trình với nhau sao cho đầu ra của chương trình này là đầu vào của một chương trình kế tiếp.



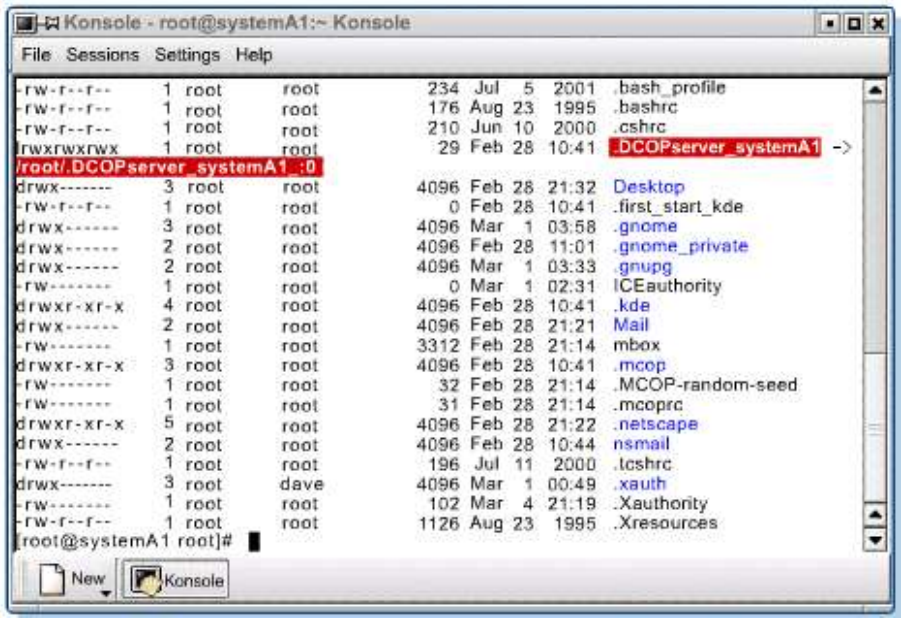
Output → Piped → Input

Named pipes cho phép truyền thông giữa hai tiến trình không quan hệ chạy trên một máy.

SHELL

- Giao tiếp người dùng sử dụng đồ họa (GUI)
- Giao tiếp dòng lệnh (CLI)

Giao tiếp người dùng đồ họa (GUI)


Konsole - root@systemA1:~ Konsole
File Sessions Settings Help
-rw-r--r-- 1 root root 234 Jul 5 2001 .bash_profile
-rw-r--r-- 1 root root 176 Aug 23 1995 .bashrc
-rw-r--r-- 1 root root 210 Jun 10 2000 .cshrc
lrwxrwxrwx 1 root root 29 Feb 28 10:41 .DCOPserver_systemA1 ->
/root/.DCOPserver_systemA1_:0
drwx------ 3 root root 4096 Feb 28 21:32 Desktop
-rw-r--r-- 1 root root 0 Feb 28 10:41 .first_start_kde
drwx------ 3 root root 4096 Mar 1 03:58 .gnome
drwx------ 2 root root 4096 Feb 28 11:01 .gnome_private
drwx------ 2 root root 4096 Mar 1 03:33 .gnupg
-rw------- 1 root root 0 Mar 1 02:31 ICEauthority
drwxr-xr-x 4 root root 4096 Feb 28 10:41 .kde
drwx------ 2 root root 4096 Feb 28 21:21 Mail
-rw------- 1 root root 3312 Feb 28 21:14 mbox
drwxr-xr-x 3 root root 4096 Feb 28 10:41 .mcop
-rw------- 1 root root 32 Feb 28 21:14 .MCOP-random-seed
-rw------- 1 root root 31 Feb 28 21:14 .mcoprc
drwxr-xr-x 5 root root 4096 Feb 28 21:22 .netscape
drwx------ 2 root root 4096 Feb 28 10:44 nsmail
-rw-r--r-- 1 root root 196 Jul 11 2000 .tcshrc
drwx------ 3 root dave 4096 Mar 1 00:49 .xauth
-rw------- 1 root root 102 Mar 4 21:19 .Xauthority
-rw-r--r-- 1 root root 1126 Aug 23 1995 .Xresources
[root@systemA1 root]#
New
Konsole

- "X Windows" cho phép Linux thao tác tương tự các GUI khác.
- Giao tiếp X Windows chuẩn giống Windows 2000 .
- Người sử dụng Linux có thể tùy biến giao tiếp X Window để giải quyết các như cầu đặc tả .
- Một cửa sổ thiết bị đầu cuối hiển thị lời nhắc lệnh Linux chuẩn trên màn hình.

Giao tiếp dòng lệnh (CLI)

```
[root@systemA1 root]# su - student A1
[studentA10systemA1 studentA1]$
[studentA10systemA1 studentA1]$ whoami
studentA1
[studentA1@systemA1 studentA1]$
```

- Giao tiếp dòng lệnh cho phép người dùng tương tác với hệ thống theo cùng cách thức lệnh thông dịch trong Windows 2000 .
- Người sử dụng có thể đưa vào các lệnh cơ bản.
- Không đưa vào các lệnh tùy tiện có thể ảnh hưởng đến hệ thống.

1. Using cd with an absolute pathname:

```
$ cd /home
```

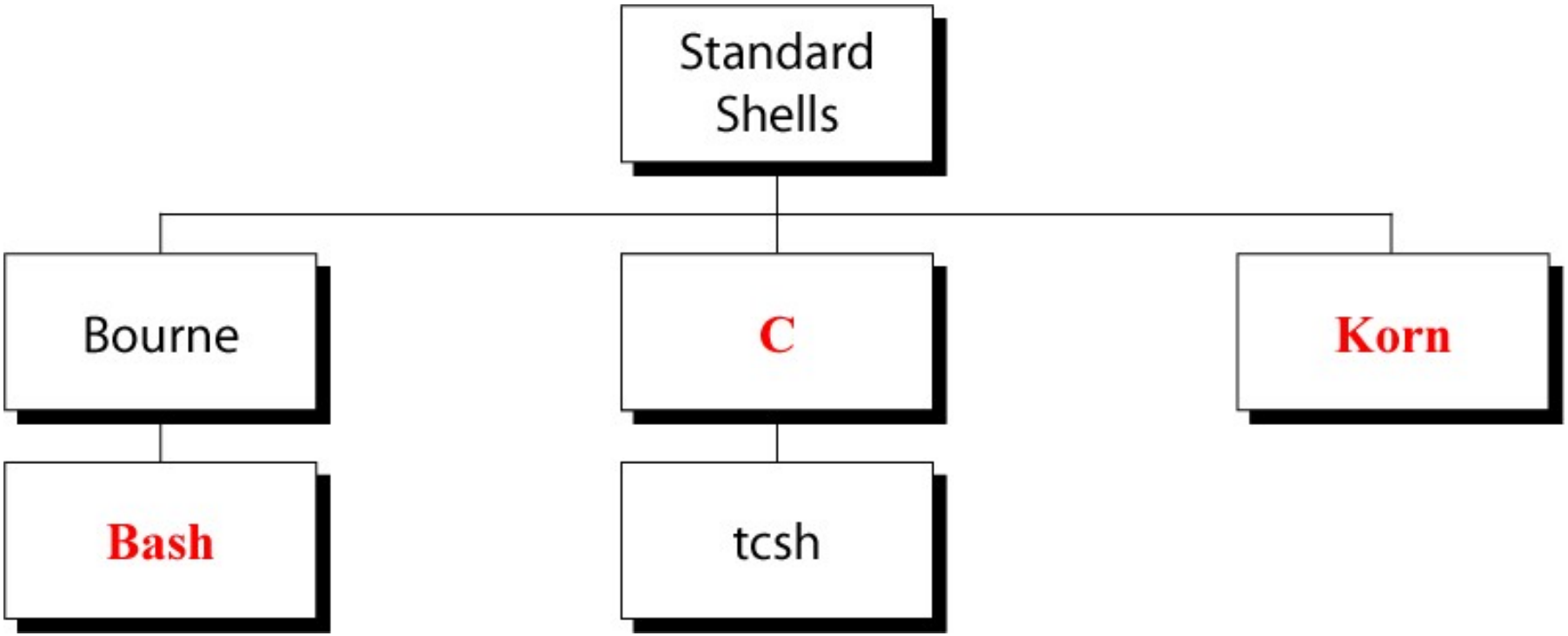
2. Using cd with a relative pathname:

```
$ cd user2/dir1
```

3. Use cd without a directory name returns you to your home directory:

```
$cd
```

Tổng quan Shell UNIX chuẩn



Các lời nhắc CLI

- **Lời nhắc** là văn bản xác định ở đó ta có thể đưa một lệnh.
- Bourne: $
- Korn: $
- Bash: $
- C Shell: %
- # được dùng khi đăng nhập như root.

The Linux Shells

```
[studentA1@systemA1 studentA1]$ ps
  PID TTY          TIME CMD
 3914 pts/1    00:00:00 bash
 3914 pts/1    00:00:00 ps
[studentA1@systemA1 studentA1]$ echo $SHELL
/bin/bash
$ ksh
[studentA1@systemA1 ~]$ps
  PID TTY          TIME CMD
 3914 pts/1    00:00:00 bash
 4162 pts/1    00:00:00 ksh
 4167 pts/1    00:00;00 ps
$
```

- Shells làm việc như một diễn dịch dòng lênh.
- Tương tự môi trường MS-DOS.
- Phổ biến nhất là *bash* shell, tổ hợp các đặc điểm tương tác làm C shell trở nên phổ biến với việc dễ sử dụng cú pháp lập trình shell của Boure shell .
- Born Again Shell (*bash* shell) được dùng cho nhiều hệ thống dòng UNIX.

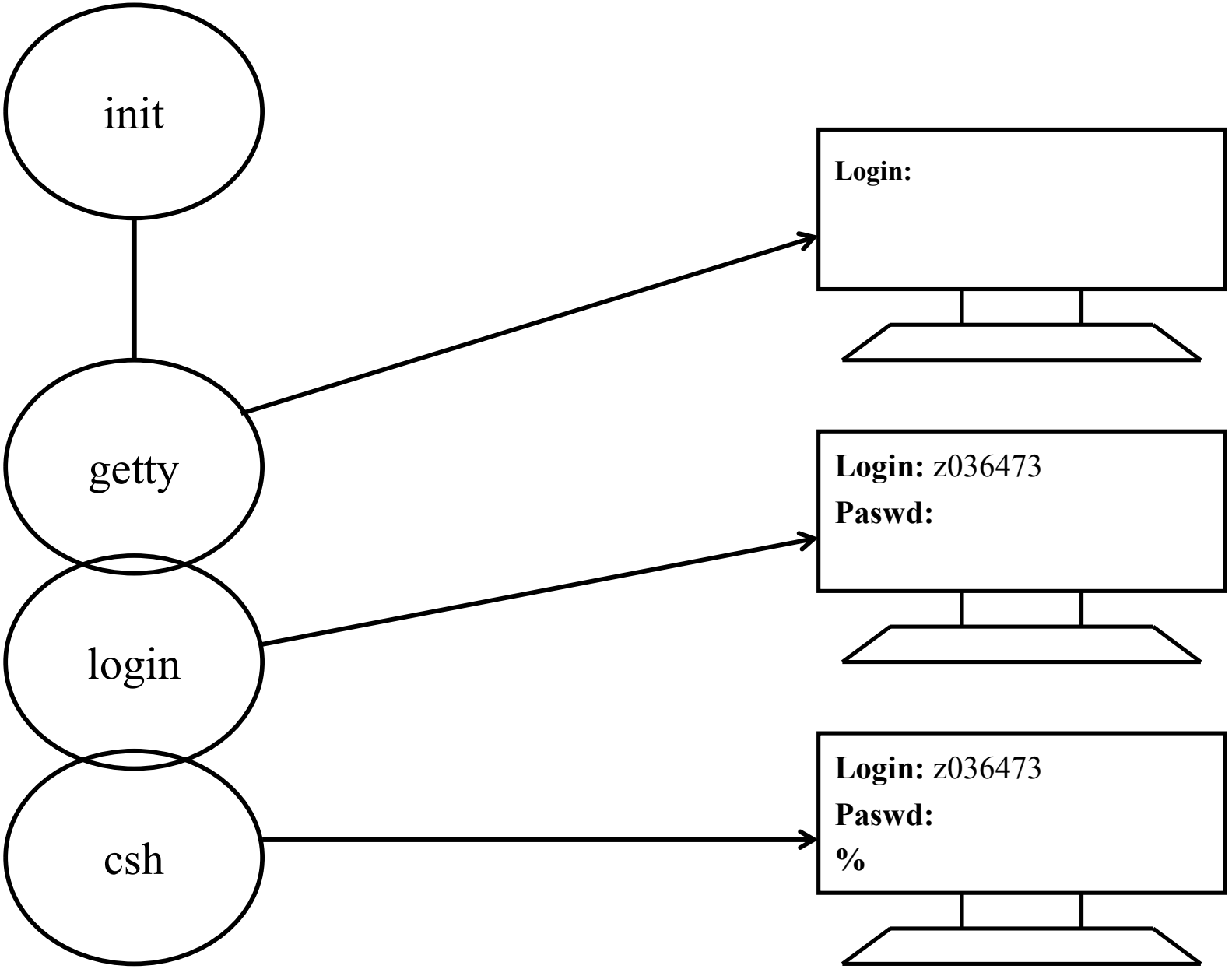
Các tiện ích của shell (bash)

- Hoàn thành tự động câu lệnh với phím <tab>
  - Cho tên câu lệnh
    - `$ema<tab> ==> $emacs`
  - Cho tham số là tên tệp
    - `$cd /usr/inc<tab> ==> $cd /usr/include`
- Gọi lại các câu lệnh trong quá khứ
  - Sử dụng phím lên () và xuống (↓) tìm các câu lệnh đã gõ trong quá khứ

C Shell : Mục đích

- Môi trường C Shell
  - Các file khởi động và đăng xuất
  - Các biến Shell
  - Lịch sử
  - Các lệnh thay thế
  - Dãy các lệnh
  - Gán bí danh
  - Chồng thư mục
  - Định hướng lại và đường ống
- Lập trình C Shell

Khởi tạo C Shell



Các công cụ cơ bản

- Các trình thông dịch lệnh (shell) : sh, csh, bash
- Các câu lệnh quản lý hệ thống tệp
- Các câu lệnh quản lý tiến trình
- Các câu lệnh xử lý dữ liệu
- Các trình soạn thảo: vi, emacs, …
- Các trình quản lý gói dữ liệu: tar, gzip,…
- Các trình biên dịch : C, C++, Fortran, Perl
- Các bộ xử lý văn bản (latex), hình ảnh (xv)
- v.v.

GIỚI THIỆU
HỆ ĐIỀU HÀNH UNIX

1. Tổng quan
2. Các thành phần hệ điều hành UNIX
3. Một số thao tác cơ bản

MỘT SỐ THAO TÁC CƠ BẢN

Đăng nhập và kết thúc

- Đăng nhập
- Đăng xuất
- Lệnh thay đổi mật khẩu ***paswd***

Lệnh liên quan đến thời gian hệ thống

a. ***cal***

b. date

Lênh liên quan đến người làm việc trên mạng

a. ***who***

b. finger :

Đăng nhập và kết thúc

a. Đăng nhập

- Mỗi người sử dụng phải sở hữu một tên đăng nhập và có một mật khẩu kèm theo
- Người sử dụng có thể đăng nhập hệ thống với tên và mật khẩu thông qua thiết bị giao tiếp (console)
- Có hai dạng console
  - Chế độ dòng lệnh (sử dụng trình thông dịch lệnh)
  - Chế độ đồ hoạ (sử dụng giao diện cửa sổ)
- Mỗi lần đăng nhập tạo ra một phiên làm việc. Phiên được kết thúc bằng câu lệnh exit hoặc logout

Đăng nhập ở chế độ dòng lệnh

- Một trình thông dịch lệnh được tự động khởi động khi phiên làm việc bắt đầu
  - Cho phép tạo tương tác với người sử dụng thông qua câu lệnh
  - Nhập lệnh bằng bàn phím, kết quả in ra dạng văn bản trên màn hình
  - Sử dụng rất ít tài nguyên nên phù hợp ngay cả khi cần tương tác từ xa
  - Hoạt động dựa trên một ngôn ngữ lập trình dạng kịch bản (script)

Console ảo

- Một phương tiện cho phép mở đồng thời nhiều phiên làm việc trên cùng một trạm làm việc
- LINUX hỗ trợ 8 console ảo trên một máy tính. Mỗi console quản lý tương ứng một phiên làm việc. Để chọn console ảo cần sử dụng tổ hợp phím tắt Ctrl+Alt+F1 cho đến F8

  Ctrl+Alt-F1 : Console ảo 1

  Ctrl+Alt-F2 : Console ảo 2

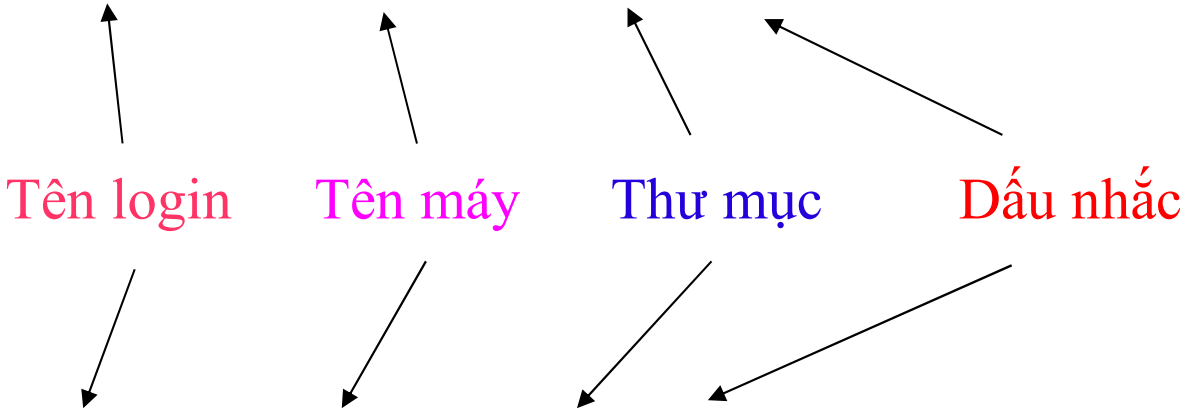
  ...

  Ctrl+Alt-F7 : Console ảo 7 (cho chế độ đồ hoạ)

Dấu nhắc câu lệnh

- Trình thông dịch lệnh (còn gọi là shell) thực hiện một cách lặp nhiều lần các công việc sau đây
  - Hiển thị dấu nhắc chờ lệnh của người sử dụng
  - Bắt câu lệnh của người sử dụng
  - Phân tích lệnh
  - Thực hiện lệnh

[student@server home]$ <lệnh của NSD thông thường>



[root@server home]# <lệnh của root>

Ví dụ đăng nhập và đăng xuất

```
login: tuananh
pasword: xxxxx
[student@server student]$ echo hello↵
hello
[student@server student]$ exit↵
```

Câu lệnh Unix

command [-options] [arguments]

- **command** là tên câu lệnh
- **options** biểu diễn một hoặc một vài tuỳ chọn
- **arguments** là các tham số câu lệnh
- Một tuỳ chọn được thể hiện bởi một kí tự đi sau dấu gạch ngang (“-”)
- Có thể nối nhiều tuỳ chọn sau một dấu gạch ngang. Ví dụ, **-asli** tương đương với **-a -s -l -i**
- Nếu một lựa chọn cần tham số kèm theo thì chúng được phân cách nhau bởi một dấu trắng
- Cần phân biệt chữ hoa và thường trong câu lệnh

Giới thiệu câu lệnh căn bản

- logname : hiên thị tên NSD đang ở phiên làm việc
- hostname : hiển thị tên trạm làm việc
- clear : xoá màn hình
- who : tên của những người đang đăng nhập
- exit : kết thúc phiên làm việc
- paswd : thay đổi mật khẩu
- date : hiển thị ngày hệ thống
- mkdir : tạo thư mục
- rmdir : xoá thư mục
- cd : chuyển vị trí thư mục
- pwd : đường dẫn thư mục hiện tại
- cp : sao chép tệp
- rm : xoá tệp
- ps : xem tiến trình
- v.v…

Trợ giúp (man)

- Xem trợ giúp trực tuyến bằng lệnh man để biết cách sử dụng các câu lệnh, cấu trúc tệp dữ liệu và cả các hàm thư viện Unix
  - `$man ls` : đặc tả câu lệnh ls
  - `$man fstab` : đặc tả tệp `/etc/fstab`
  - `$man fgetc` : đặc tả hàm thư viện `fgetc`
- Sử dụng các phím sau để xem trợ giúp
  - <enter> tiến một dòng
  - <space> tiến một trang
  - <b> quay lại một trang
  - <q> thoát khỏi trợ giúp

Lệnh thời gian hệ thống

a. Lệnh cal

- Chức năng: Hiện lịch theo tháng hay năm
- Cú pháp: **cal {tham số}**

tham số: [[tháng] năm]

Lệnh thời gian hệ thống(tiếp)

b. Lệnh date

- Chức năng: Hiện hay thay đổi ngày giờ hệ thống
- Cú pháp:

date [tùy chọn]... [+Khuôn dạng]

Tùy chọn -d -s --help

Khuôn dạng:

%a thứ viết tắt (Sun..Sat)
%A thứ viết đầy đủ (Sunday..Saturday)
%b Tháng viết tắt (Jan..Dec)
%B Tháng viết đầy đủ (January..December)
%c Ngày địa phương (Sat Nov 04 12:02:33 EST 1989)
%C năm trong thế kỷ (năm chia dư 100) [00-99]
%d ngày trong tháng (01..31)
%D (mm/dd/yy)

Lệnh liên quan đến người dùng trên mạng

a. Lệnh who

- Chức năng: Liệt kê các thông tin người dùng trên mạng
- Cú pháp **who [tùy chọn]**
  Tùy chọn:

-a mọi thông tin

-b thời gian khởi động hệ thống gần nhất

-d in các tiến trình bị hủy

-m đưa thông tin máy và người đang sử dụng

-p các tiến trình hoạt động

-q tên và số lượng người dùng

-u danh sách người dùng

Lệnh liên quan đến người dùng trên mạng (tiếp)

b. Lệnh finger

- Chức năng: Hiển thị thông tin về những người dùng hệ thống
- Cú pháp:

finger[-lmsp] [user] [user@host]

Tùy chọn:

-s hiển thị tên, tên thực, tên thiết bị đầu cuối, thời điểm đăng nhập, số phone

-l Hiển thị nội dung các file trong thư mục người dùng

-m Tránh trùng tên (nhạy chữ hoa)

-p Hạn chế tác dụng tùy chọn -l

BÀI TẬP

1.Đang nhập ở chế độ GUI và trở ra.

2. Đăng nhập ở chế độ CLI

3. Thay đổi mật khẩu, đăng nhập lại, khôi phục mật khẩu

4. Viết lệnh hiện lịch tháng hiện thời, lịch cả năm 2000

5. Viết lệnh hiện thông tin về người dùng yêu cầu trên mạng

6. Viết lệnh liệt kê người dùng trên mạng

7. Chuyển sang màn hình làm việc khác, đăng nhập với tên người dùng khác

8. Xem trợ giúp trực túyên đối với lệnh chmod

9. Kết thúc phiên làm việc